

EXHIBIT C-3

Artec Studio/CR Studio on disk project comparison

Yet another interesting detail about CR Studio is that it uses almost identical project structure compared to Artec Studio 10 project. Let's see the differences and similarities.

Overview

Let's compare main parameters of the projects saved in Artec Studio 10 and CR Studio / MagicWand Studio

	Artec Studio	CR Studio
Structure of the project	Main file with .sproj and data files in folders	Main file with .scanproj and data files in folders
Format of the project file	xml	xml
Name of the folder with data	Scans	Scans

Main file

Now let's compare main project file content from Artec Studio and from CR Studio with the same (1) number of scans.



We see almost identical structure of the file with the same fields, their names and their meanings.

Let's compare line by line:

Line 1 - standard xml header

Line 2 - "ArtecProject" renamed to "RulerProject", fields "Version" and "Name" has the same value and meaning.

Line 3 - <Scans Count="1"> - identical, same value and meaning

Line 4 - <Scan ... /> - description of first scan:

- fields "Uuid", "Name", "Items", "Size_kb" - have the same values and meaning
- fields "LoadMode", "Selected", "DataType" - have the same meaning, notation slightly differ

- fields “**HasSlammed**”, “**AliasName**” - introduced by Crealitiy
- it’s worth to mention that Uuid is a main field in this line identifying a set of files containing 3D data stored in “**Scans**” folder

Line 5 - closing bracket from line 3

Line 6 - closing bracket from line 2

So we see that majority of all fields in the project along with their values and meanings in CR Studio are identical to Artec Studio.

Meaning and purpose of fields is very special for Artec Software. Please find description from Artec code database:

sproj - xml-файл with below structure

- ArtecProject – parent top level node
 - Attributes
 - Version - version of format of project file
 - Name – project name
 - Uuid - unique project identifier
 - Rev – project revision (increases each time you save project)
 - Daughter nodes
 - Scans - container that stores information about project scans
 - Attributes
 - Count – number of scans in the project
 - Daughter nodes
 - Scan - contains description of one scan
 - Attributes
 - Uuid - unique scan identifier (scan with UUID identifier is saved in the Scans directory with the name UUID.dscan/UUID.mscan)
 - Name - name of the scan - it is shown in the Workspace window in Artec Studio
 - Items - number of frames in the scan
 - !LoadMode - scan loading mode. Possible values are "scan", "nothing", "carcas".
 - Selected - ("true"/"false") - whether the scan is selected in the Workspace window (marked with an eye)
 - Size_kb - approximate size of a fully loaded scan in memory (in kilobytes)
 - DataType - data type. It duplicates the same flag in the .meta file for the .dscan file, this is done to more efficiently read properties when loading a project
 - 0 - normal scan
 - 1 - fusion

Scans

In both applications 3D data are stored in folder named “**Scans**”. Each object is represented by several files with the name set to “**Uuid**” from the main file (sproj or scanproj correspondingly) with different extensions. Let’s see how they are stored

Artec Studio project scans folder with file types .mscan, .dscan, and .tscan

> ... > 49 Brown Side Table (Small) > Scans ▾ 👤

Name ↑

📄	{0D4DEC84-AB67-41BD-8B5D-52A5AB35FE75}.dscan	👤
📄	{0D4DEC84-AB67-41BD-8B5D-52A5AB35FE75}.mscan	👤
📄	{0D4DEC84-AB67-41BD-8B5D-52A5AB35FE75}.tscan	👤
📄	{8C27F3E6-F425-4671-8BCF-42F1A4249607}.dscan	👤

CRStudio project Scans folder with file types .mscan, .rscan, and .scanbuf

> ... > test2 > Scans ▾ 👤

Name ↑

📄	{7ef9aeb9-5a58-4a08-8ce7-e2f1ab470ebb}.mscan	👤
📄	{7ef9aeb9-5a58-4a08-8ce7-e2f1ab470ebb}.rscan	👤
📄	{472213ea-3e11-4c79-a8b9-1d8ed1a21c14}.rscan	👤
📄	{472213ea-3e11-4c79-a8b9-1d8ed1a21c14}.scanbuf	👤
📄	{e7f826c4-c97d-4930-870c-57e8d65c2e03}.rscan	👤

Files in this folder is binary and has different format but structure is the same: to open the project one need to read <scan .. /> line from main project file, get “Uuid” field and then open and read files from the “Scan” folder with the name set to “Uuid” and specific extensions (different in each application).

History of Project structure creation in Artec

Important note that such a discrepancy in the field names - a mixture of SnakeCase and CamelCase naming styles - usually occurs when different people come up with names and no one follows the general concept, when the code does not pass the code review. And it was exactly the situation when the code was developed at the beginning of Artec Studio creation. Therefore, it is almost impossible to meet the same names in another application and also called them with a different styles.

Fields - Uuid, Name, Items, LoadMode, Selected, Size_kb – was created by Artec employees before 2011. Below screenshot from Artec code database shows their existence for more than 11 years.

```

1294     }
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314     {
1315         wxXmlNode* sc_node = new wxXmlNode(wxXML_ELEMENT_NODE, wxT("Scan"));
1316         sc_node->AddProperty(wxT("Uuid"), scan->getScanUuid());
1317         sc_node->AddProperty(wxT("Name"), scan->getTitle());
1318         sc_node->AddProperty(wxT("Items"), wxString::Format(wxT("%d"), scan->size()));
1319         sc_node->AddProperty(wxT("LoadMode"), loadedMode(scan->determineLoadedState()));
1320         sc_node->AddProperty(wxT("Selected"), scan->isSelected() ? wxT("true") : wxT("false"));
1321         sc_node->AddProperty(wxT("Size_kb"), wxString::Format(wxT("%d"), scan->getScanFullMemorySize()/1024));
1322         scans_node->AddChild(sc_node);
1323     }
1324 }
1325 scans_node->AddProperty(wxT("Count"), wxString::Format(wxT("%d"), scans_count));

```

Final thoughts

It's hard to imagine that Creality without looking into Artec Studio independently created identical structure and format of the project by chance creating from the scratch even such weird names as "**Size_kb**". It is also hard to imagine that they were inspired by our project structure because there is nothing special about it, it's not implemented in any open source library and is not fully compatible as binary files differs.

The most reasonable explanation is that Creality just had Artec Source code that implemented this functionality and used it.